**Code demo:**



**1. HTML:**

HTML (HyperText Markup Language) là ngôn ngữ để xây dựng cấu trúc cho trang web. Một trang HTML cơ bản bao gồm các thẻ (tags), được sử dụng để định nghĩa các phần tử khác nhau trên trang. Dưới đây là giải thích chi tiết hơn về một số thẻ HTML quan trọng:

1. **Thẻ <!DOCTYPE>**:
   * <!DOCTYPE html> khai báo loại tài liệu là HTML5. Đây là phiên bản HTML phổ biến nhất hiện nay và là tiêu chuẩn được sử dụng trên hầu hết các trang web.

**Ví dụ**:

<!DOCTYPE html>  
<html lang="en">  
<head>

1. **Thẻ <html>**:
   * Thẻ <html> bọc toàn bộ nội dung của trang web. Nó báo cho trình duyệt biết rằng nội dung bên trong là tài liệu HTML.

**Ví dụ**:

html

<!DOCTYPE html>  
<html lang="en">  
<!-- Nội dung của trang HTML sẽ nằm ở đây -->  
</html>

1. **Thẻ <head>**:
   * Thẻ <head> chứa thông tin về tài liệu HTML, bao gồm tiêu đề trang, liên kết đến các tệp CSS, JavaScript và các thông tin meta (chẳng hạn như thông tin về bộ mã hóa ký tự, viewport,...).
   * Nội dung trong thẻ <head> không hiển thị trực tiếp trên trang web.

**Ví dụ**:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Tiêu đề trang</title>  
  
</head>  
  
</html>

1. **Thẻ <body>**:
   * Thẻ <body> chứa toàn bộ nội dung sẽ hiển thị trên trang web, bao gồm văn bản, hình ảnh, bảng biểu, form,...

**Ví dụ**:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>HTML, CSS, JavaScript Example</title>  
 <link rel="stylesheet" href="styles.css">  
 <script src="script.js"></script>  
</head>  
<body>  
<h1>Tiêu đề lớn</h1>  
<p>Đoạn văn bản hiển thị trên trang web.</p>  
  
</body>  
</html>

1. **Thẻ <h1> đến <h6>**:
   * Đây là các thẻ tiêu đề (heading). Thẻ <h1> là tiêu đề lớn nhất, còn <h6> là tiêu đề nhỏ nhất.

**Ví dụ**:

<h1>Tiêu đề chính</h1>  
<p>Đây là một đoạn văn bản được hiển thị trên trang web.</p>  
  
<div>  
 <h2>Nội dung bên trong một div</h2>  
 <p>Đoạn văn bản trong thẻ div.</p>  
</div>

1. **Thẻ <p>**:
   * Thẻ <p> dùng để định nghĩa một đoạn văn bản (paragraph).

**Ví dụ**:

<p>Đoạn văn bản trong thẻ div.</p>

1. **Thẻ <a>**:
   * Thẻ <a> dùng để tạo liên kết (link) đến một URL khác.

**Ví dụ**:

html

<a href="https://www.example.com">Nhấp vào đây để đến trang khác</a>

1. **Thẻ <div>**:
   * Thẻ <div> là một thẻ dùng để chứa các phần tử khác và thường được sử dụng để tạo bố cục (layout) cho trang web.

**Ví dụ**:

html

<div>  
 <h2>Nội dung bên trong một div</h2>  
 <p>Đoạn văn bản trong thẻ div.</p>  
</div>

1. **Thẻ <input> và <button>**:
   * Thẻ <input> là phần tử nhập liệu của người dùng. Loại nhập liệu được xác định bằng thuộc tính type, ví dụ text, password, date, v.v.
   * Thẻ <button> dùng để tạo nút bấm.

**Ví dụ**:

<input type="text" id="input-text" placeholder="Nhập tên của bạn">  
<button onclick="sayHello()">Nhấn vào đây</button>

**2. CSS:**

CSS (Cascading Style Sheets) được dùng để định nghĩa giao diện của các phần tử HTML như màu sắc, kích thước, vị trí, v.v. Có ba cách chính để nhúng CSS vào một tài liệu HTML:

1. **Nhúng CSS trực tiếp (Inline CSS)**: Bạn có thể định nghĩa CSS ngay trong thẻ HTML bằng cách sử dụng thuộc tính style.

**Ví dụ**:

html

<p style="color: red; font-size: 16px;">Đoạn văn bản này có màu đỏ và kích thước chữ 16px.</p>

1. **Nhúng CSS trong thẻ <style> (Internal CSS)**: Bạn có thể đặt các định nghĩa CSS trong thẻ <style> nằm trong thẻ <head> của tài liệu HTML.

**Ví dụ**:

html

<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>HTML, CSS, JavaScript Example</title>  
 <link rel="stylesheet" href="styles.css">  
 <script src="script.js"></script>  
 <style>  
 p {  
 color: blue;  
 font-size: 18px;  
 }  
 </style>  
</head>

1. **Liên kết tệp CSS bên ngoài (External CSS)**: Đây là cách thông dụng nhất, bạn liên kết tệp CSS bên ngoài bằng thẻ <link>.

**Ví dụ**:

html

<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>HTML, CSS, JavaScript Example</title>  
 <link rel="stylesheet" href="styles.css">  
</head>

**Một số thuộc tính CSS cơ bản:**

1. **color:** Đặt màu cho văn bản.

**Ví dụ**:

p {  
 color: blue;  
 font-size: 18px;  
}

1. **font-size:** Đặt kích thước chữ.

**Ví dụ**:

css

h1 {  
 font-size: 32px;  
}

1. **background-color:** Đặt màu nền cho một phần tử.

**Ví dụ**:

div {  
 background-color: lightgray;  
}

1. **padding:** Đặt khoảng cách bên trong giữa nội dung và biên của phần tử.

**Ví dụ**:

div {  
 padding: 10px;  
}

1. **margin:** Đặt khoảng cách giữa phần tử và các phần tử khác.

**Ví dụ**:

div {  
 background-color: lightgray;  
 margin: 20px;  
 padding: 10px;  
}

1. **border:** Đặt đường viền cho phần tử.

**Ví dụ**:

div {  
 background-color: lightgray;  
 margin: 20px;  
 padding: 10px;  
 border: 2px solid black;  
}

**3. JavaScript:**

JavaScript là ngôn ngữ lập trình giúp trang web có thể tương tác với người dùng một cách linh hoạt. JS thường được sử dụng để thay đổi nội dung trang theo thời gian thực, xử lý sự kiện từ người dùng, hoặc thực hiện các tính toán phức tạp.

**Cách nhúng JavaScript vào HTML:**

1. **Nhúng trực tiếp (Inline JavaScript):** Bạn có thể viết JavaScript trực tiếp trong một thẻ HTML bằng cách sử dụng thuộc tính onclick, onchange,...

**Ví dụ**:

<button onclick="alert('Xin chào!')">Nhấn vào đây</button>

1. **Nhúng trong thẻ <script> (Internal JavaScript):** Bạn có thể viết mã JavaScript trong thẻ <script> nằm ở phần <head> hoặc cuối phần <body>.

**Ví dụ**:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <script>  
 function sayHello() {  
 alert('Chào mừng bạn đến với trang web!');  
 }  
 </script>  
</head>  
<body>  
 <button onclick="sayHello()">Nhấn vào đây</button>  
</body>  
</html>

1. **Liên kết tệp JavaScript bên ngoài (External JavaScript):** Bạn có thể viết mã JavaScript trong một tệp riêng và liên kết nó với HTML bằng thẻ <script>.

**Ví dụ**:

<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>HTML, CSS, JavaScript Example</title>  
 <link rel="stylesheet" href="styles.css">  
 <script src="script.js"></script>  
</head>

**Trong tệp script.js**:

function sayHello() {  
 alert("Chào mừng bạn đến với trang web!");  
}

**Một số khái niệm cơ bản về JavaScript:**

1. **Biến (Variables):** Biến là nơi lưu trữ giá trị. Bạn có thể khai báo biến bằng từ khóa let, const hoặc var.

**Ví dụ**:

let *name* = "John";  
const *age* = 30;

1. **Hàm (Functions):** Hàm là khối mã có thể thực thi một nhiệm vụ nào đó.

**Ví dụ**:

function greet() {  
 alert("Hello, World!");  
}

1. **Điều kiện (Conditionals):** JavaScript cho phép bạn kiểm tra điều kiện với câu lệnh if.

**Ví dụ**:

javascript

let *age2* = 20;  
if (*age2* >= 18) {  
 alert("Bạn đã đủ tuổi.");  
}

1. **Vòng lặp (Loops):** Vòng lặp cho phép bạn thực hiện một tác vụ lặp đi lặp lại nhiều lần.

**Ví dụ**:

javascript

for (let i = 0; i < 5; i++) {  
 *console*.log("Lần lặp thứ " + i);  
}

1. **Tương tác với DOM:** Bạn có thể sử dụng JavaScript để truy cập và thay đổi nội dung của các phần tử HTML trên trang web thông qua DOM (Document Object Model).

**Ví dụ**:

javascript

*document*.getElementById('myElement').innerHTML = "Nội dung mới";

**Ví dụ về jQuery trong JavaScript:**

jQuery là một thư viện JavaScript giúp đơn giản hóa việc thao tác với DOM và xử lý sự kiện.

**Ví dụ sử dụng jQuery từ file claim-manager.js**:

javascript

$('#search-button').click(function() {  
 alert("Nút tìm kiếm đã được nhấn!");  
});

* $('#search-button'): Truy cập đến phần tử có ID là search-button.
* .click(function() {...}): Lắng nghe sự kiện "click" và thực thi hàm bên trong khi nút được nhấn.

**Kết luận:**

* **HTML**: Cung cấp cấu trúc cơ bản của trang web với các thẻ như <html>, <head>, <body>, <input>, và <button>.
* **CSS**: Định nghĩa giao diện cho trang web bằng cách sử dụng các thuộc tính như color, font-size, padding, và có thể được nhúng theo nhiều cách (inline, internal, external).
* **JavaScript**: Cung cấp khả năng tương tác với người dùng, thay đổi nội dung trang web và xử lý sự kiện. Bạn có thể nhúng JavaScript trực tiếp hoặc liên kết từ tệp bên ngoài.